NAIST Simultaneous Speech-to-Text Translation System for IWSLT 2022

Ryo Fukuda[†], **Yuka Ko**[†], **Yasumasa Kano**[†], **Kosuke Doi**[†], **Hirotaka Tokuyama**[†], **Sakriani Sakti**[†‡], **Katsuhito Sudoh**[†], **Satoshi Nakamura**[†]
[†]Nara Institute of Science and Technology, Japan
[‡]Japan Advanced Institute of Science and Technology, Japan
fukuda.ryo.fo3@is.naist.jp

Abstract

This paper describes NAIST's simultaneous speech translation systems developed for IWSLT 2022 Evaluation Campaign. We participated the speech-to-speech track for English-to-German and English-to-Japanese. Our primary submissions were end-to-end systems using adaptive segmentation policies based on Prefix Alignment.

1 Introduction

This paper describes NAIST's submissions to IWSLT 2022 (Anastasopoulos et al., 2022) Simultaneous Speech Translation track. We participated the speech-to-speech track for English-to-German (En-De) and English-to-Japanese (En-Ja) using our end-to-end simultaneous machine translation (SimulMT) systems.

SimulMT based on neural machine translation (NMT) has achieved a large success in recent years. There are two different SimulMT approaches depending on the policy that determines READ (waiting for speech input) and WRITE (writing text output) actions: *fixed* and *adaptive*. Fixed policies are usually implemented by simple rules (Dalvi et al., 2018; Ma et al., 2019; Fukuda et al., 2021; Sen et al., 2021). They are simple yet often effective, but they sometimes make inappropriate decisions due to large word order differences, pauses, and so on. In contrast, adaptive policies decide READ or WRITE actions flexibly taking current context into account (Zheng et al., 2019a,b, 2020; Liu et al., 2021). They can be more effective than fixed policies in end-to-end speech-to-speech SimulMT because it is difficult to define fixed policies for speech input.

In our systems, we use Bilingual Prefix Alignment (Kano et al., 2022), which extracts alignment between bilingual prefix pairs in the training time, for prefix-to-prefix translation in SimulMT. The Bilingual Prefix Alignment is applied to extract prefix pairs of source language speech and target language translations. We also use the prefix pairs to train a boundary prediction model for an adaptive speech segmentation policy. Our system showed some improvements against wait-k baselines on the development data, in all the latency regimes in both En-De and En-Ja.

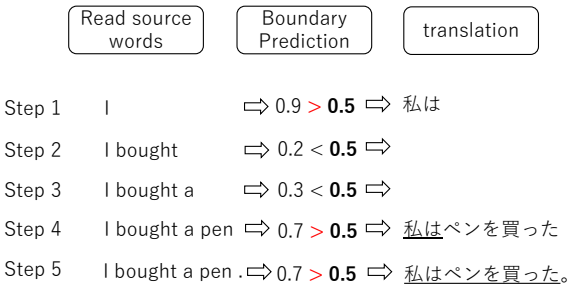

Figure 1: A brief overview of our prefix-to-prefix translation process (Kano et al., 2022) from English to Japanese. The threshold of boundary probability is 0.5 in this example. Underlined parts are the forced output prefixes.

2 Simultaneous Speech Translation based on Bilingual Prefix Alignment

We developed simultaneous speech translation (SimulST) based on offline speech translation (ST). Our SimulST system translates an incrementally-growing source language speech prefix into the target language. When the system detects a segment boundary in source language speech, the latest segment is translated taking its input and translation history into account. The ST model is basically the same as an offline one, and we used it to translate an input prefix speech segment from the beginning. However, we constrained the translation prefix by the results in the previous time step. The constraint is implemented by a forced decoding with a given translation prefix. Figure 1 shows an example of whole translation process, but we input the speech prefixes with fixed number of frames. Please refer

Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022), pages 286 - 292

to (Kano et al., 2022) for details of Bilingual Prefix Alignment.

For this system, we need an ST model using an ST corpus consisting of source language speech segments and corresponding translations in the target language. We then fine-tune the offline ST model with prefix pairs of source language speech and target language translations obtained using Bilingual Prefix Alignment. We also need a boundary predictor to segment source language speech adaptively as SimulMT policies. In this section, we present how to extract prefix pairs (2.1) and build the boundary predictor (2.2).

2.1 Extracting Prefix Pairs

Suppose we already have an offline ST model trained using an ST corpus and are going to extract prefix pairs for a speech segment in the source language (S). First, we extract the speech prefixes with τ, 2τ, 3τ, ... frames. Then, for each speech prefix S_{prefix}, we translate it into $\hat{T}_{prefix}$ using the offline ST model. Finally, we compare $\hat{T}_{prefix}$ with $\hat{T}_{offline}$, which is a translation of the entire speech segment. If $\hat{T}_{prefix}$ appears as a prefix of $\hat{T}_{offline}$, we extact $(S_{prefix}, \hat{T}_{prefix})$ as a prefix pair. We apply this process to all the source prefixes. Here, we use a forced decoding with the previously extracted prefix $\hat{T}_{prefix}$ to obtain latter prefix translations and update $\hat{T}_{offline}$ to extract consistent prefix translations. We may obtain the same target prefix with different source prefixes within a given speech segment. We just extract the first appearance and ignore the rest with longer speech prefixes in such cases. The procedure above sometimes extracts *unbalanced* prefix pairs, in which a source language prefix does not fully match its target language speech counterpart. Such unbalanced prefix pairs frequently appear between English and Japanese and cause the degradation of the translation performance. We use a simple heuristic rule to filter out them based on the length ratio between source language speech and target language translation. We exclude prefix pairs in which the length ratio len_s/len_t exceeds $maxratio$, where len_s is the length of S_{prefix} (in the number of frames) and len_t is the length of $\hat{T}_{prefix}$ (in the number of words).

2.2 Boundary Predictor

In inference, the SimulST system incrementally reads source speech and predicts a segment boundary in every τ frames.

To train the boundary predictor, we prepare pairs of a speech prefix and the corresponding binary label sequence extracted from the training data. One source language speech derives many speech prefixes in τ, 2τ, 3τ, ... frames. Suppose we extracted 2τ- and 5τ-frame speech prefixes from the same utterance, for example. We assign a label sequence with τ `0`s followed τ `1`s to the 2τ-frame prefix, which means we should predict a boundary in the second τ frames but not in the first τ frames. For the 5τ-frame prefix, we assign a label sequence where the second and fifth τ-frame parts are filled with `1`s and the rest with `0`s, consistently with the 2τ-frame prefix. In addition, we also extracted speech prefixes where the last τ-frame part is not a boundary. For example, the last τ-frame part of the 3τ- and 4τ-frame speech prefixes is filled with `0`s in this case. The boundary predictor is trained using weighted cross-entropy loss normalized in inverse proportional to the number of appearances of each label.

During inference, the boundary predictor predicts a boundary in every τ frames as a binary classification output. The prediction is made on every frames in the τ-frame segment, so we obtain τ binary classification outputs. If the proportion of label `1` here is larger than or equals to λ_{thre}, the predictor makes a decision of *boundary*, otherwise *non-boundary*.

3 Primary System

We developed SimulST systems for two language pairs: English-to-German (En-De) and English-to-Japanese (En-Ja). We implemented both our systems based on fairseq[1] (Ott et al., 2019).

3.1 End-to-end Speech Translation

3.1.1 Data

We used MuST-C v2 (Di Gangi et al., 2019), a multilingual ST corpus extracted from TED talks subtitles. Each dataset consists of triplets of segmented English speech, transcripts, and target language translations. The En-De and En-Ja datasets contained about 250k and 330k segments, respectively. As acoustic features, we used 80-dimensional log Mel filter bank (FBANK) with global-level cepstral mean and variance normalization (CMVN) applied.

[1] https://github.com/pytorch/fairseq/commit/acf312418e4718996a103d67bd57516938137a7d

We applied with Byte Pair Encoding (BPE) to split the sentences into subwords using SentencePiece (Kudo and Richardson, 2018), with a vocabulary of 20,000 subwords shared across the source and target languages.

3.1.2 Model

We used the Transformer implementation of fairseq to build the models. We trained the ASR model using the English speech-text pairs and then trained the ST model using the ASR model for the parameter initialization. The architecture of ASR and ST models were the same. The encoder consisted of a 2D-convolution layer that reduces the sequence length to a quarter, and 12 transformer encoder layers. The decoder consisted of six transformer decoder layers. We set the embedding dimensions and the feed-forward dimensions to 256 and 2,048 and used four attention heads for both the encoder and decoder. The model was trained using Adam with an initial learning rate of 0.0005 with warmup updates of 10,000. In the En-De ASR and ST models and the En-Ja ASR model, we performed the dropout probability of 0.1 and set early stopping patience to 16. In the En-Ja ST model, we set the dropout probability of 0.2 and set early stopping patience to 32.

The ST model training was in two steps. We first trained the ST model using entire segment pairs from the MuST-C. We then fine-tuned the model using bilingual prefix pairs extracted using Bilingual Prefix Alignment (2.1).

3.1.3 Evaluation

We evaluated the models with BLEU and Average Lagging (AL) (Ma et al., 2019) using SimulEval (Ma et al., 2020) on MuST-C v2 tst-COMMON. For En-De, we evaluated on the best ST model based on the dev set, and for En-Ja, we evaluated on the checkpoint averaged ST model in last 10 epochs. Our proposed models were decoded with beam search (beam size=10).

3.2 Implementation Details of the Proposed Method

3.2.1 Data Extraction

We extracted training data for the ST model and the boundary prediction model by using Bilingual Prefix Alignment described in section 2. We set $\tau = 100$ and tried $maxratio = \{\text{None}, 80, 40, 20\}$.

System	BLEU	AL
Offline	21.04	-
Baseline		
wait-1	3.66	844.45
wait-5	11.49	1684.13
wait-17	18.80	3786.07
Proposed (λ_{thre})		
low (0.1)†	17.54	990.32
medium (0.47)	19.15	1859.56
high (0.68)	19.50	3896.67

Table 1: The main results of our systems on En-De tst-COMMON. † uses $T = 48$ frames as an input unit.

System	BLEU	AL
Offline	11.6	-
Baseline		
wait-7	4.76	2369.68
wait-17	8.46	3723.65
wait-27	9.55	4421.75
Proposed (λ_{thre})		
low (0.0)	9.26	2185.51
medium (0.36)	9.90	3946.02
high (0.4)	10.22	4733.65

Table 2: The main results of our systems on En-Ja tst-COMMON. The FT model was the best model with data filtering approach.

3.2.2 Boundary Predictor

We trained the boundary predictor using the extracted source language speech prefixes. The boundary predictor consisted of a 2D-convolution layer reducing the sequence length to $\tau/4$ (25 frames), a unidirectional LSTM layer, and an output linear layer that gives label probabilities $\hat{x}_n \in R^2$ at the n-th frame of the convolution layer. We set the embedding dimensions and the hidden state dimensions of the LSTM layer to 256 and 512. The model was trained using Adam with an initial learning rate of 0.0001, warmup updates of 4,000 and early stopping patience of 8. During inference, we tried several values of voting threshold λ_{thre} between 0.0 to 1.0 to adjust for latency and BLEU tradeoffs.

4 Experiments

We conducted comparative experiments with wait-k (Ma et al., 2019). For baseline wait-k, we tried k ranging from 1 to 19 at two intervals for En-De and 5 to 31 at two intervals (excluding 29) for En-Ja.

Metrics	En-De	En-Ja
Accuracy	0.678	0.679
Precision	0.646	0.480
Recall	0.490	0.009
F1	0.557	0.017

Table 3: The evaluation results of boundary predictor models on prefix pairs of tst-COMMON dataset in $\lambda_{thre} = 0.5$.

Following the default wait-k setting in fairseq, one unit for k was set to 280 frames. For examples, when $k = 3$, after reading 3×280 frames, the model would WRITE and READ alternately.

4.1 Main Results

Table 1 shows the best results of the proposed and baseline SimulMT systems in En-De with low (AL $\leq$ 1,000), medium (AL $\leq$ 2,000), and high (AL $\leq$ 4,000) latency regimes. Table 2 shows the counterpart in En-Ja with low (AL $\leq$ 2,500), medium (AL $\leq$ 4,000), and high (AL $\leq$ 5,000) latency regimes. In both language pairs, our model outperformed the baselines with all the latency regimes. In particular, the proposed method showed a significant improvement of more than 10 points in BLEU in En-De with low latency regime. On the other hand, the improvement for En-Ja was smaller than in En-De. One possible reason was the performance difference of the boundary predictor, which depends on the difference between source and target languages. Table 3 shows the results of the boundary predictor on prefix pairs of tst-COMMON dataset with $\lambda_{thre} = 0.5$. For both language pairs, the accuracy was under 68%, suggesting the difficulty of binary classification at the acoustic frame level. Especially, the recall of En-Ja boundary predictor was extremely low, which means that its output predictions were almost 0 (READ) in $\lambda_{thre} = 0.5$. The small λ_{thre} value was required to output label 1 (WRITE) frequently on En-Ja, compared to En-De, as shown in Tables 1 and 2.

4.2 Effectiveness of Fine-tuning

Figure 2 shows the results of wait-k baselines, a model fine-tuned with bilingual prefix pairs (FT) and a model without fine-tuning (w/o FT). Figure 3 shows the counterparts in En-Ja. In En-De, the fine-tuned model worked better than the non fine-tuned model in the range of AL $\leq$ 4,000. The performance gap between proposed models and wait-k models in the low latency ranges were larger than

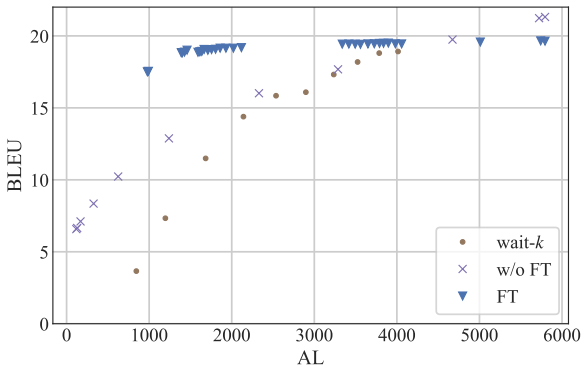

Figure 2: The BLEU and AL results of FT, w/o FT and baseline in En-De. The two FT points in low latency regime (AL$\leq$1000) were evaluated in $T = 48$ frames on $\lambda_{thre} = \{0.0, 0.1\}$.

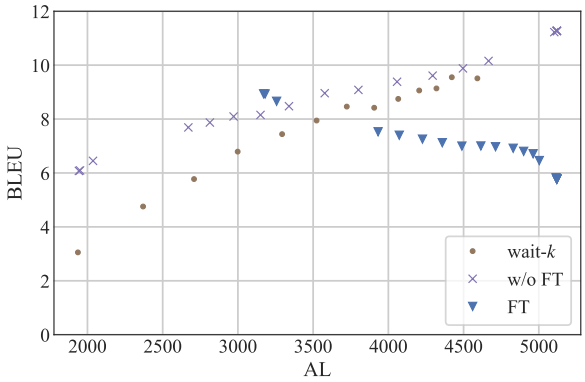

Figure 3: The BLEU and AL results of FT, w/o FT and baseline in En-Ja. The FT model was fine-tuned with non-filtered prefix pairs.

those in the high latency ranges. On the other hand, the non-fine-tuned model worked better than the fine-tuned model in the very large latency ranges with AL $>$ 4000. Both of them outperformed the baseline wait-k models consistently in BLEU. The fine-tuned model achieved higher BLEU scores at the cost of the larger latency, compared to the non-fine-tuned and wait-k models.

In En-Ja, the scores of the non-fine-tuned model were better than those of wait-k baselines with all the latency regimes. The performance improvements of the non-fine-tuned model against wait-k models in the low latency ranges were larger than those in the high latency ranges. However, the scores of the fine-tuned model were worse than those of wait-k models and the non-fine-tuned model almost everywhere. It suggests the failure of appropriate fine-tuning in En-Ja.

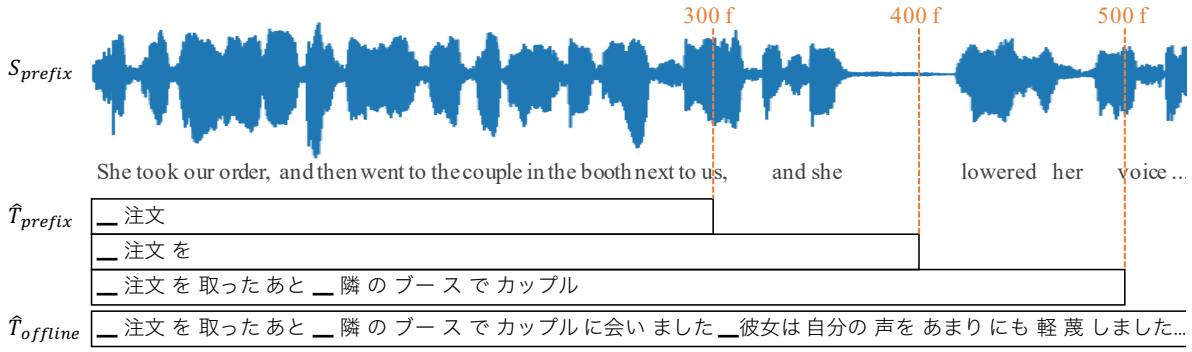

Figure 4: Examples of extracted prefix pairs on En-Ja containing unbalanced pairs whose target prefix is too short.

Filter ($maxratio$)	# samples (% removed)
None	642,426 (0%)
80	583,986 (9.1%)
40	447,517 (30.3%)
20	161,309 (74.9%)

Table 4: The samples size of En-Ja prefix alignment data filtered by $maxratio$. $maxratio$ indicates ratio between source speech frames size and target hypothesis tokens length.

	Offline (hyp/ref)
w/o FT	11.6 (0.885)
FT + Filter ($maxratio$)	
None	6.0 (0.515)
80	6.4 (0.530)
40	8.0 (0.609)
20	10.9 (0.796)

Table 5: The En-Ja FT BLEU results on offline with filtered prefix alignment data. hyp/ref indicates ratio between hypothesis length and reference length.

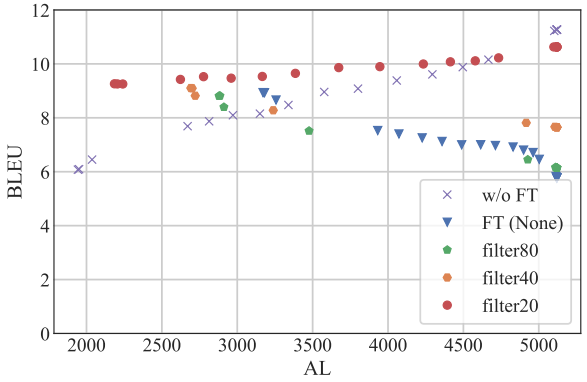

Figure 5: The En-Ja BLEU and AL results of w/o FT models and FT models. The FT models were fine-tuned with filtered prefix alignment data.

4.2.1 Data Filtering for English-Japanese

In contrast to En-De, the fine-tuned model was inferior to the non-fine-tuned and wait-k models in En-Ja. We expected that under-translation would degrade the performance because the fine-tuning used prefix pairs of a long source language speech prefix and a short target language text segment. It would be due to differences in sentence structures between English and Japanese. Since English and German are subject-verb-object (SVO) languages, the English prefix speech frames and the German prefix tokens can be aligned without long-distance reordering. For example, the pair dataset of English frames and German tokens {English prefix frames, German prefix tokens} would consist of {S, S}, {SV, SV}, {SVO, SVO}. On the other hand, since Japanese is a subject-object-verb (SOV) language, the difference in sentence structures between them causes the difficulty in aligning prefixes. For example, the prefix pairs of English speech and Japanese text {English prefix frames, Japanese prefix tokens} would consist of {S, S}, {SV, S}, {SVO, SOV}. Such an unbalanced pair like {SV, S} would make the fine-tuned model prefer inappropriately short outputs. Figure 4 shows examples of prefix pairs extracted using Bilingual Prefix Alignment to fine-tune the ST model. Bilingual Prefix Alignment extracted unbalanced pairs $(S_{prefix}, \hat{T}_{prefix})$ whose target prefix is too short. For example, a source speech prefix of 300 frames (about three seconds) is paired with a target prefix of only two subwords, which obviously does not match.

We applied simple data filtering described in 2.1 for En-Ja. Table 4 shows the prefix alignment dataset with the filtering. The filtering can reduce the unbalanced pairs of data that consists of long source speech frames and short target tokens. It

would alleviate the model to generate too short sequences. Table 5 shows the results of the fine-tuned model with the filtered prefix pairs. Table 5 shows the BLEU improvement from no filter setting (None) to larger $maxratio$ filter setting with alleviating the gap between hypothesis length and reference length (*hyp/ref*). Figure 5 shows the results of the fine-tuned (FT) models with filtered prefix alignment dataset. FT (None) was worse than the non-fine-tuned model in the latency ranges with AL $>$ 3500. The scores by the fine-tuned model using filtered data on $maxratio = 80$ (filter80) were almost the same as FT (None) model's. Decreasing $maxratio$ to 20 significantly improved BLEU scores. It suggests selective use of the fine-tuning data alleviated the under-translation problem for distant language pairs.

5 Conclusions

In this paper, we described our SimulST systems in English-to-German and English-to-Japanese. The proposed method uses prefix alignment data to fine-tune the offline ST model and train boundary predictor that judges when to READ and WRITE. Our models achieved some improvements compared to the wait-k baselines in every latency regime in both English-to-German and English-to-Japanese.

Acknowledgement

Part of this work was supported by JSPS KAKENHI Grant Number JP21H05054.

References

Antonios Anastasopoulos, Luisa Bentivogli, Marcely Z. Boito, Ondřej Bojar, Roldano Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, Marcello Federico, Christian Federmann, Hongyu Gong, Roman Grundkiewicz, Barry Haddow, Benjamin Hsu, Dávid Javorský, Věra Kloudová, Surafel M. Lakew, Xutai Ma, Prashant Mathur, Paul McNamee, Kenton Murray, Maria Nădejde, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, Juan Pino, Elizabeth Salesky, Jiatong Shi, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Yogesh Virkar, Alex Waibel, Changhan Wang, and Shinji Watanabe. 2022. FINDINGS OF THE IWSLT 2022 EVALUATION CAMPAIGN. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, Dublin, Ireland. Association for Computational Linguistics.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, and Stephan Vogel. 2018. Incremental decoding and training methods for simultaneous translation in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 493–499, New Orleans, Louisiana. Association for Computational Linguistics.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

Ryo Fukuda, Yui Oka, Yasumasa Kano, Yuki Yano, Yuka Ko, Hirotaka Tokuyama, Kosuke Doi, Sakriani Sakti, Katsuhito Sudoh, and Satoshi Nakamura. 2021. NAIST English-to-Japanese simultaneous translation system for IWSLT 2021 simultaneous text-to-text task. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 39–45, Bangkok, Thailand (online). Association for Computational Linguistics.

Yasumasa Kano, Katsuhito Sudoh, and Satoshi Nakamura. 2022. Simultaneous neural machine translation with prefix alignment. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, Dublin, Ireland. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP (Demonstration)*.

Dan Liu, Mengge Du, Xiaoxi Li, Ya Li, and Enhong Chen. 2021. Cross attention augmented transducer networks for simultaneous translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 39–55, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036, Florence, Italy. Association for Computational Linguistics.

Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. 2020. SIMULEVAL: An evaluation toolkit for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 144–150, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Sukanta Sen, Ulrich Germann, and Barry Haddow. 2021. The University of Edinburgh's submission to the IWSLT21 simultaneous translation task. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 46–51, Bangkok, Thailand (online). Association for Computational Linguistics.

Baigong Zheng, Kaibo Liu, Renjie Zheng, Mingbo Ma, Hairong Liu, and Liang Huang. 2020. Simultaneous translation policies: From fixed to adaptive. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2847–2853, Online. Association for Computational Linguistics.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019a. Simpler and faster learning of adaptive policies for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1349–1354, Hong Kong, China. Association for Computational Linguistics.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019b. Simultaneous translation with flexible policy via restricted imitation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5816–5822, Florence, Italy. Association for Computational Linguistics.